Arithmetic Operations in the Polynomial Modular Number System

Jean-Claude Bajard
Laurent Imbert
Thomas Plantard

LIRMM, Universite Montpellier II, France
ATIPS, CISaC, University of Calgary, Canada

28 june 2005

Plan

Introduction

Arithmetic needs for public key cryptography

Diffie-Hellman

- An exponentiation over the prime field F_p
- Needs : Multiplication modulo p (prime)
- Length : $1024, 2048, \ldots$ bits

RSA

- An exponentiation on the ring $\mathbb{Z}/n\mathbb{Z}$
- Needs : Multiplication modulo n (composite, $n = p.q$)
- Length : $1024, 2048, \ldots$ bits

ECC

- Elliptic curve point multiplication
- Needs : Arithmetic operations (+,-,*,/) over the finite field F_q, where q is a power of a prime p
- Length : $160, 192, \ldots$ bits

New Number System

Number system

Positional number system with radix β

$$X = \sum_{i=0}^{n-1} x_i \beta^i \qquad \text{with } x_i \in \{0, ...\beta - 1\}$$

Example : $X = 1315 = (3,4,4,2)_8 = 3 + 4 \times 8 + 4 \times 8^2 + 2 \times 8^3$

Modular number system **MNS**(p, n, γ, ρ)

$$X = \sum_{i=0}^{n-1} x_i \gamma^i \bmod P \qquad \text{with } x_i \in \{0, \ldots, \rho - 1\}$$

Example

- $MNS(p = 17, n = 3, \gamma = 7, \rho = 3)$
- $a = \sum_{i=0}^{2} x_i 7^i$ mod 17 with $a_i \in \{0, 1, 2\}$

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	

Example

- $MNS(p = 17, n = 3, \gamma = 7, \rho = 3)$
- $a = \sum_{i=0}^{2} x_i 7^i \bmod 17$ with $a_i \in \{0, 1, 2\}$

0	1	2	3	4	5
$(0,0,0)$	$(0,0,1)$	$(0,0,2)$			

6	7	8	9	10	11

12	13	14	15	16

Example

- $MNS(p = 17, n = 3, \gamma = 7, \rho = 3)$
- $a = \sum_{i=0}^{2} x_i 7^i \bmod 17$ with $a_i \in \{0, 1, 2\}$

0	1	2	3	4	5
$(0,0,0)$	$(0,0,1)$	$(0,0,2)$			
6	7	8	9	10	11
	$(0,1,0)$	$(0,1,1)$	$(0,1,2)$		
12	13	14	15	16	

Example

- $MNS(p = 17, n = 3, \gamma = 7, \rho = 3)$
- $a = \sum_{i=0}^{2} x_i 7^i \bmod 17$ with $a_i \in \{0, 1, 2\}$

0	1	2	3	4	5
$(0,0,0)$	$(0,0,1)$	$(0,0,2)$			
6	7	8	9	10	11
	$(0,1,0)$	$(0,1,1)$	$(0,1,2)$		
12	13	14	15	16	
		$(0,2,0)$	$(0,2,1)$	$(0,2,2)$	

Example

- $MNS(p = 17, n = 3, \gamma = 7, \rho = 3)$
- $a = \sum_{i=0}^{2} x_i 7^i$ mod 17 with $a_i \in \{0, 1, 2\}$

0	1	2	3	4	5
$(0, 0, 0)$	$(0, 0, 1)$	$(0, 0, 2)$			$(1, 1, 0)$
6	7	8	9	10	11
$(1, 1, 1)$	$(0, 1, 0)$	$(0, 1, 1)$	$(0, 1, 2)$		
12	13	14	15	16	
$(1, 2, 0)$	$(1, 2, 1)$	$(0, 2, 0)$	$(0, 2, 1)$	$(0, 2, 2)$	

Example

- $MNS(p = 17, n = 3, \gamma = 7, \rho = 3)$
- $a = \sum_{i=0}^{2} x_i 7^i \bmod 17$ with $a_i \in \{0, 1, 2\}$

0	1	2	3	4	5
$(0,0,0)$	$(0,0,1)$	$(0,0,2)$	$(2,1,1)$	$(2,1,2)$	$(1,1,0)$
6	7	8	9	10	11
$(1,1,1)$	$(0,1,0)$	$(0,1,1)$	$(0,1,2)$	$(2,2,0)$	$(2,2,1)$
12	13	14	15	16	
$(1,2,0)$	$(1,2,1)$	$(0,2,0)$	$(0,2,1)$	$(0,2,2)$	

How find a “good” Modular Number System ?

What do we need ?

1. A MNS where ρ is small (about $\rho \sim p^{1/n}$)
2. A “fast” arithmetic on the MNS

Definition : AMNS

A modular number system $\mathcal{B} = MNS(p, n, \gamma, \rho)$ is called Adapted Modular Number System (AMNS) if

$$\gamma^n \bmod P = c,$$

with c is a small integer.

Fundamental Theorem

Fundamental Theorem

Definition

A $MNS(p, n, \gamma, \rho)$ is called Polynomial Modular Number System (PMNS) if $\exists E(X) = X^n + aX + b$ such that

1. E is irreducible in $\mathbb{Z}[X]$
2. $E(\gamma) \equiv 0 \pmod{p}$
3. $\rho \geq (|a| + |b|)p^{1/n}$

Theorem

A $PMNS$ can represent all the integer of $[0, p-1]$.
$\forall a \in [0, p-1], \exists A \in \mathbb{Z}[X]$ such that

1. $A(\gamma) = a \bmod p$
2. $\deg A < n$
3. $\|A\|_\infty = \max_{0 \leq i < n}\{|a_i|\} < \rho$

Remark

1. Proof use Lattice Theory ($\sim CVP_\infty$)
2. Algorithmic solution is long : Babai...

Example

Example

1. We choose $p = 250043$
2. We choose $n = 3$
3. We have $X^3 - 2$ is irreducible in $\mathbb{Z}[X]$.
4. We have $\gamma = 127006$ is a root of $X^3 - 2$ modulo p

ρ

1. $(|0| + |-2|)p^{1/3} = 2.250043^{1/3} < 128 = \rho$
2. $PMNS(p = 250043, n = 3, \gamma = 127006, \rho = 128)$

Arithmetic on PMNS

Arithmetic on PMNS

Modular Multiplication in PMNS

1. Polynomial multiplication in $\mathbb{Z}[X]$: $C(X) \leftarrow A(X)B(X)$
2. Polynomial reduction : $C'(X) \leftarrow C(X) \bmod E(X)$
3. Coefficient reduction : $R \leftarrow CR(V)$, gives $R(\gamma) \equiv C'(\gamma) \pmod p$

Generalization

operation on PMNS $\rightarrow$ polynomial operation + coefficient reduction

Example

PMNS

$PMNS(p = 250043, n = 3, \gamma = 127006, \rho = 128)$

Input

- $A = 7 + 30X + 100X^2 \Rightarrow A = 65842$
- $B = 59 + 2X + 76X^2 \Rightarrow B = 8816$

Algorithm

1. $C(X) = A(X) \times B(X)$
   $U(X) = 413 + 1784X + 6492X^2 + 2480X^3 + 7600X^4$
2. $C'(X) = C(X) \bmod (X^3 - 2) \leftarrow 5373 + 16984X + 6492X^2$
3. $R(X) = ?$

General Coefficient Reduction

Input

- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$

General Coefficient Reduction

Input

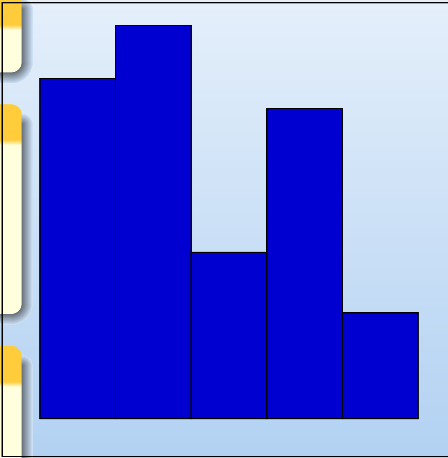
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



General Coefficient Reduction

Input

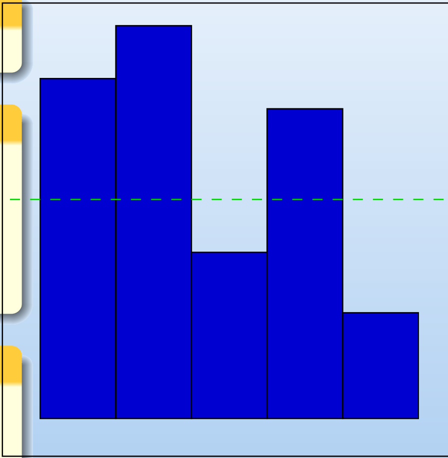
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



General Coefficient Reduction

Input

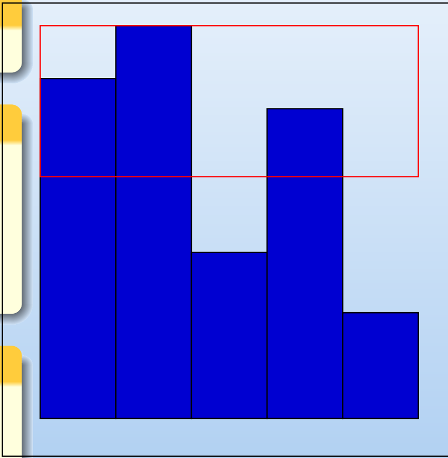
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



General Coefficient Reduction

Input

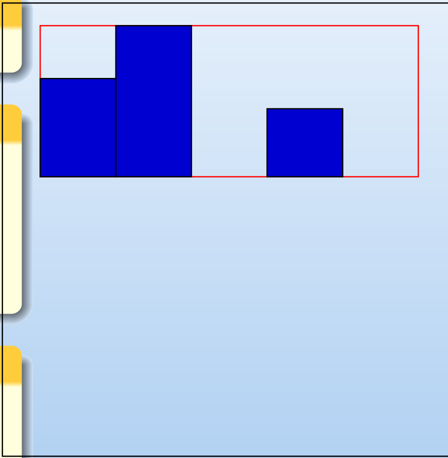
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



General Coefficient Reduction

Input

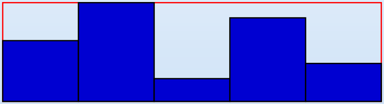
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



General Coefficient Reduction

Input

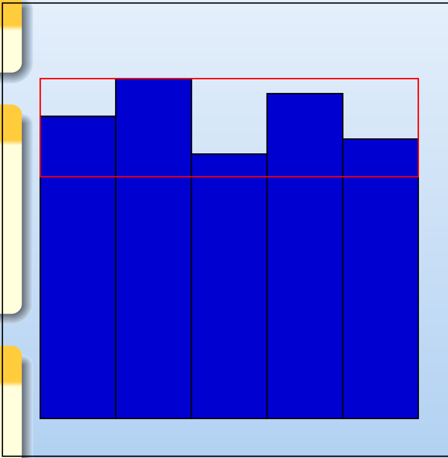
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



General Coefficient Reduction

Input

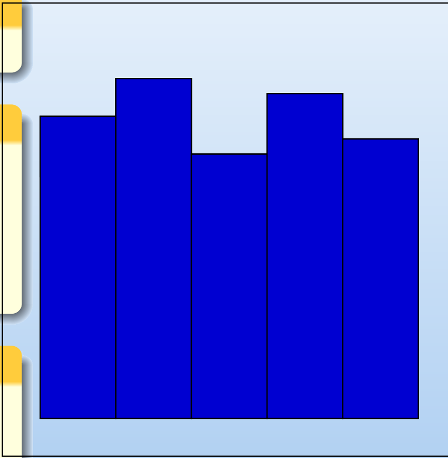
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



General Coefficient Reduction

Input

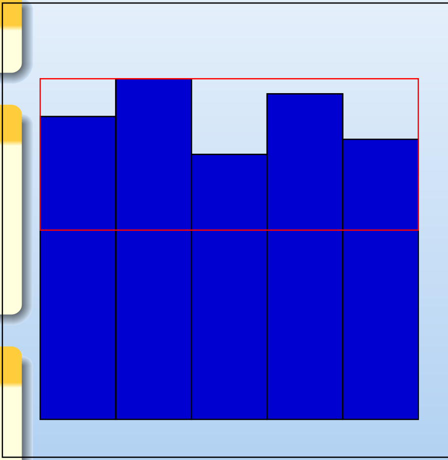
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



General Coefficient Reduction

Input

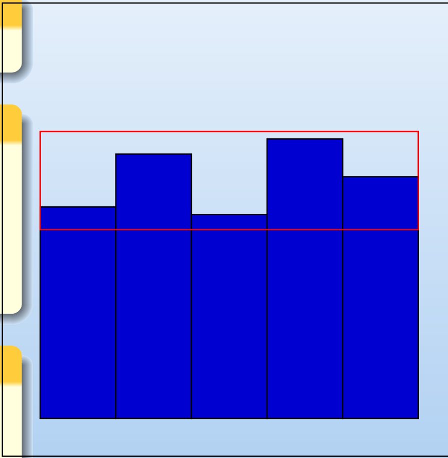
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



General Coefficient Reduction

Input

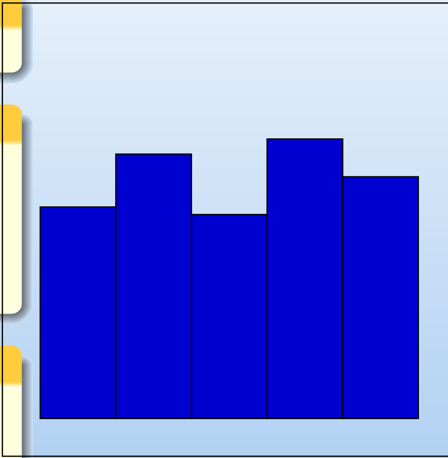
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



General Coefficient Reduction

Input

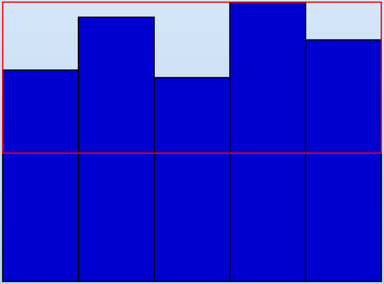
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



General Coefficient Reduction

Input

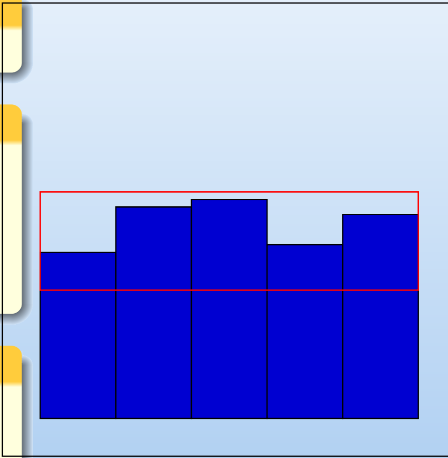
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



General Coefficient Reduction

Input

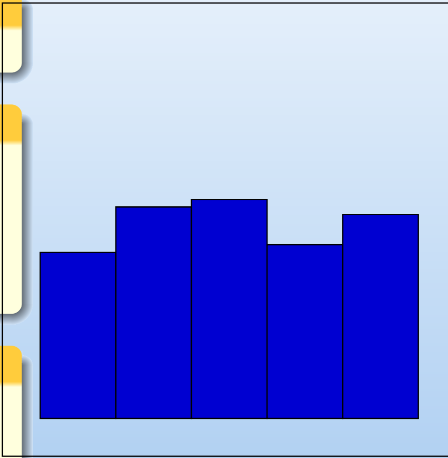
- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$

Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$



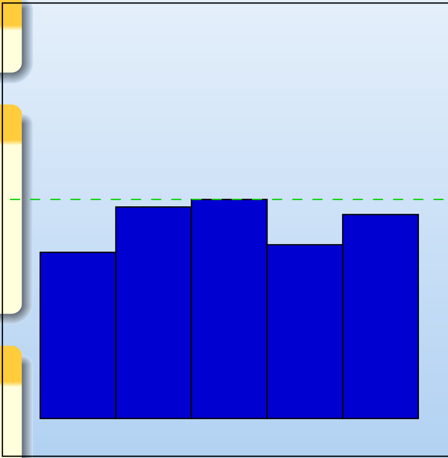
General Coefficient Reduction

Input

- A vector V with $\|V\|_\infty < 2^t$

Algorithm

1. $R \leftarrow V$
2. WHILE $t > k_s$ DO
   1. $R = \overline{R}2^{t-k_e} + \underline{R}$
   2. $\overline{R} \leftarrow RED(\overline{R})$
   3. $R \leftarrow \overline{R}2^{t-k_e} + \underline{R}$
   4. $t \leftarrow t - (k_e - k_s)$



Output

- A vector $R \equiv V$
- With coefficients $\|R\|_\infty < \rho = 2^{k_s}$

The RED algorithm

Input

- A vector V with $\|V\|_\infty < 2^{k_e}$

Algorithm

1. $V = U 2^{k_s - 1} + L$
2. $U \leftarrow Table(U)$
3. $R \leftarrow U + L$

Output

A vector $R \equiv V$ with $\|R\|_\infty < 2^{k_s}$

The RED algorithm

Input

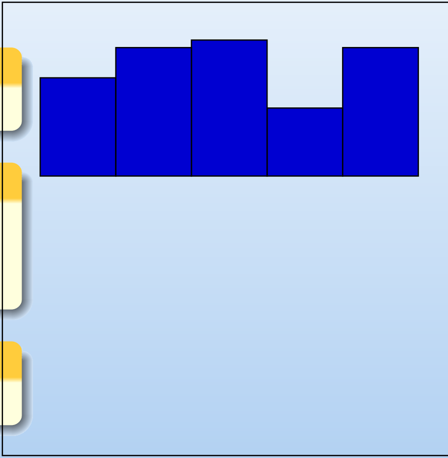
- A vector V with $\|V\|_\infty < 2^{k_e}$

Algorithm

1. $V = U2^{k_s-1} + L$
2. $U \leftarrow Table(U)$
3. $R \leftarrow U + L$

Output

A vector $R \equiv V$ with $\|R\|_\infty < 2^{k_s}$



The RED algorithm

Input

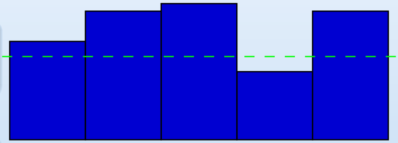
- A vector V with $\|V\|_\infty < 2^{k_e}$

Algorithm

1. $V = U2^{k_s-1} + L$
2. $U \leftarrow Table(U)$
3. $R \leftarrow U + L$

Output

A vector $R \equiv V$ with $\|R\|_\infty < 2^{k_s}$



The RED algorithm

Input

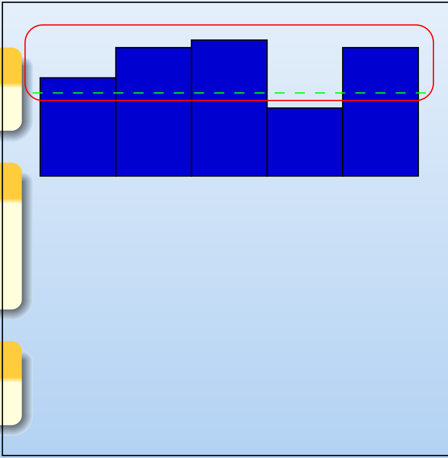
- A vector V with $\|V\|_\infty < 2^{k_e}$

Algorithm

1. $V = U2^{k_s-1} + L$
2. $U \leftarrow Table(U)$
3. $R \leftarrow U + L$

Output

A vector $R \equiv V$ with $\|R\|_\infty < 2^{k_s}$



The RED algorithm

Input

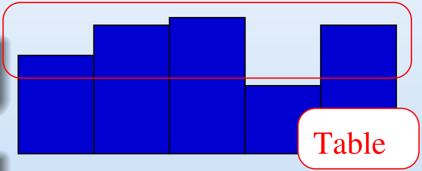
- A vector V with $\|V\|_\infty < 2^{k_e}$

Algorithm

1. $V = U2^{k_s - 1} + L$
2. $U \leftarrow Table(U)$
3. $R \leftarrow U + L$

Output

A vector $R \equiv V$ with $\|R\|_\infty < 2^{k_s}$



The RED algorithm

Input

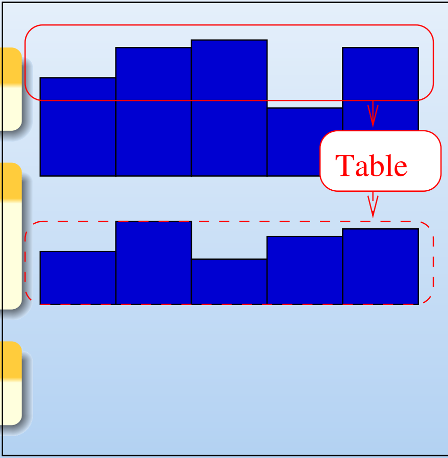
- A vector V with $\|V\|_\infty < 2^{k_e}$

Algorithm

1. $V = U2^{k_s - 1} + L$
2. $U \leftarrow Table(U)$
3. $R \leftarrow U + L$

Output

A vector $R \equiv V$ with $\|R\|_\infty < 2^{k_s}$



The RED algorithm

Input

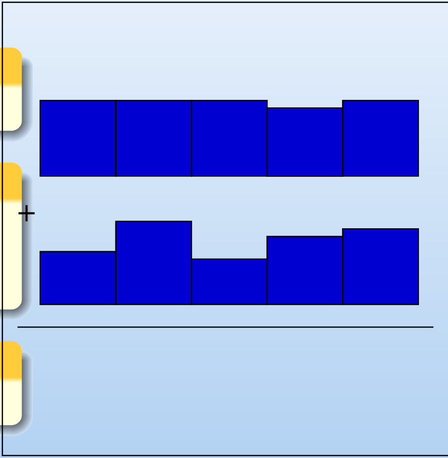
- A vector V with $\|V\|_\infty < 2^{k_e}$

Algorithm

1. $V = U2^{k_s - 1} + L$
2. $U \leftarrow Table(U)$
3. $R \leftarrow U + L$

Output

A vector $R \equiv V$ with $\|R\|_\infty < 2^{k_s}$



The RED algorithm

Input

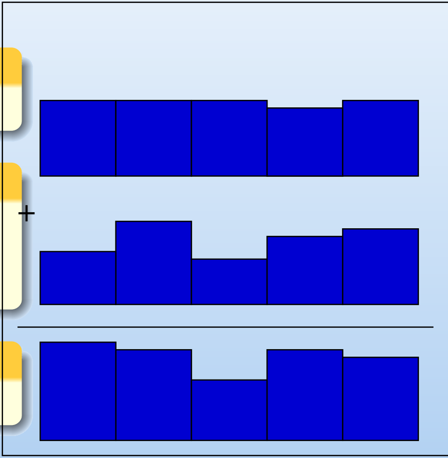
- A vector V with $\|V\|_\infty < 2^{k_e}$

Algorithm

1. $V = U2^{k_s - 1} + L$
2. $U \leftarrow Table(U)$
3. $R \leftarrow U + L$

Output

A vector $R \equiv V$ with $\|R\|_\infty < 2^{k_s}$



The RED algorithm

Input

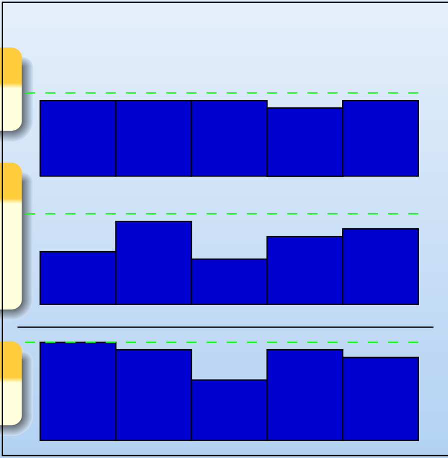
- A vector V with $\|V\|_\infty < 2^{k_e}$

Algorithm

1. $V = U2^{k_s - 1} + L$
2. $U \leftarrow Table(U)$
3. $R \leftarrow U + L$

Output

A vector $R \equiv V$ with $\|R\|_\infty < 2^{k_s}$



Conclusions

Conclusions and future directions

What we proposed

- A new number system well adapted to modular arithmetic, called **Modular Number System (MNS)**
- A theorem which allows us to define MNS having "nice" properties (small ρ)
- Table-based algorithms for the arithmetic operations (+,-,*,conversions) in the MNS

Future works

- Adapt algorithms like Montgomery and Barrett to the MNS in order to avoid table-based methods